

KROSS-PLATFORMALI DASTURLASH (FLUTTER, REACT NATIVE) ORQALI AMALIY LOYIHALAR ISHLAB CHIQUISH

To‘xtayeva Robiya Ravshanbek qizi

Andijon davlat universiteti talabasi

tukhtayeva_robiya@adu.uz

Annotatsiya: Ushbu maqolada kross-platformali dasturlash texnologiyalari, xususan Flutter va React Native platformalari asosida mobil ilovalarni ishlab chiqishning nazariy va amaliy jihatlari tahlil qilinadi. Mobil ilovalarga bo‘lgan talabning ortib borishi zamonaviy texnologik yondashuvlarni joriy etishni talab qilmoqda. Flutter va React Native’ning arxitekturasini, dizayn yondashuvlari, samaradorlik darajasi, foydalanuvchi tajribasi va amaliy loyihalar asosida ularning imkoniyatlari solishtirildi. Amaliy tajriba orqali bu texnologiyalarning o‘ziga xos afzalliklari, cheklovlari va real foydalanishdagi samaradorligi baholandi. Maqola natijalariga ko‘ra, kross-platformali yondashuvlar mobil dasturlash sohasida tejamkorlik va funksionallikni ta’minlashda muhim ahamiyat kasb etmoqda.

Kalit so‘zlar: Kross platformali dasturlash, Flutter, React Native, mobil ilovalar, amaliy loyihalar, foydalanuvchi tajriba, platformalararo interfeys, dasturiy samaradorlik.

Tadqiqotning maqsadi: Zamonaviy raqamli jamiyatda mobil ilovalarga bo‘lgan talab keskin ortib bormoqda. Bugungi kunda foydalanuvchilar turli operatsion tizimlarda ishlaydigan, qulay, funksional va tezkor ilovalarni afzal ko‘rishmoqda. Shu bois dasturchilar oldida har ikki asosiy platforma – Android va iOS uchun bir vaqtning o‘zida moslashuvchan va samarali ilovalar yaratish zarurati yuzaga kelmoqda. Mazkur ehtiyojlar kross-platformali dasturlash texnologiyalarining rivojlanishiga kuchli turtki berdi. Kross-platformali dasturlash deganda, bir martalik kod yozish orqali bir nechta platformalarda ishlay oladigan ilovalarni yaratish texnologiyasi tushuniladi. Ushbu yondashuv ishlab chiqish vaqtini qisqartirish, resurslarni tejash va ilovani tezda bozorga chiqarish imkoniyatlarini taqdim etadi. Ayniqsa, Flutter va React Native kabi ilg‘or texnologiyalar bu yo‘nalishda katta muvaffaqiyatlarga erishgan. Flutter Google tomonidan ishlab chiqilgan bo‘lib, Dart dasturlash tiliga asoslangan va vizual interfeysni yuqori sifatda qurishga imkon beradi. React Native esa Facebook tomonidan ishlab chiqilgan bo‘lib, JavaScript asosida ishlaydi va keng hamjamiyatga ega. Ushbu maqolada kross-platformali dasturlash texnologiyalarining zamonaviy ta’lim va sanoatdagi o‘rni, ularning imkoniyatlari, amaliy loyihalarda qo‘llanishi, samaradorlik mezonlari hamda foydalanuvchi tajribasiga ta’siri o‘rganiladi. Tadqiqotning maqsadi – Flutter va React Native texnologiyalari yordamida amaliy

loyihalar ishlab chiqish jarayonini ilmiy tahlil qilish, ularning afzallik va kamchiliklarini aniqlash hamda real loyihalar misolida ularning natijadorligini baholashdan iborat. Ushbu mavzuning dolzarbligi shundan iboratki, texnologik rivojlanish sharoitida mobil dasturlar ishlab chiqish samaradorligini oshirish, bir vaqtning o'zida keng platformalarda ishlaydigan ilovalarni yaratish imkonini beruvchi yondashuvlar ustuvor ahamiyat kasb etmoqda. Amaliy jihatdan esa, bu texnologiyalar o'rta va kichik bizneslardan tortib, yirik korporativ loyihalargacha keng miqyosda tatbiq etilmoqda. Shu bois maqola natijalari nafaqat nazariy, balki amaliy foyda ham keltirishi mumkin.

Kross-platformali dasturlash texnologiyalariga nazariy yondashuv zamonaviy dasturiy ta'minot ishlab chiqish jarayonida muhim o'rin egallamoqda. An'anaviy yondashuvda har bir platforma (masalan, Android va iOS) uchun alohida kod bazasi yaratilishi zarur bo'lgan bo'lsa, kross-platformali texnologiyalar bitta kod asosida bir nechta operatsion tizimlar uchun bir vaqtning o'zida dastur yaratish imkonini beradi. Bu esa, ishlab chiqish vaqtini, byudjetni va inson resurslarini tejashga olib keladi. Bugungi kunda bu yo'nalishda eng ommabop texnologiyalardan biri bu Flutter hisoblanadi. Flutter — bu Google tomonidan ishlab chiqilgan, Dart dasturlash tiliga asoslangan ochiq manbali UI toolkit bo'lib, mobil, veb va desktop ilovalar yaratishda keng qo'llaniladi. Uning asosiy arxitekturasini "widget"larga asoslanadi, ya'ni barcha interfeys komponentlari mustaqil va qayta ishlatiladigan widgetlar ko'rinishida yaratiladi. Flutter'da rendering engine (Skia) bevosita o'z grafik interfeysini quradi, bu esa unga yuqori darajadagi moslashuvchanlik va vizual aniqlik beradi. Flutter'ning afzalliklari sifatida bir xil foydalanuvchi interfeysini barcha platformalarda kafolatlash, tez prototiplash imkoniyati, keng kutubxonalar to'plami va faol hamjamiyatni keltirish mumkin. Shu bilan birga, React Native texnologiyasi ham kross-platformali ishlab chiqishda keng qo'llanilmoqda. React Native — bu Facebook tomonidan yaratilgan JavaScript asosidagi open-source framework bo'lib, unda ishlab chiqilgan ilovalar aslida mahalliy (native) komponentlardan foydalanadi. React Native'da UI komponentlari JavaScript orqali tasvirlanadi, ammo ular platformaga mos real komponentlarga aylantiriladi. Bu esa foydalanuvchiga aniq va silliq ishlaydigan interfeysni taqdim etadi. Texnologiya "Virtual DOM", "Hot Reload" va modulli arxitektura orqali ishlab chiqish jarayonini sezilarli darajada tezlashtiradi. Shu bilan birga, mavjud JavaScript bilimlari bo'lgan dasturchilar uchun React Native'ni o'zlashtirish osonroq hisoblanadi. Flutter va React Native o'rtasidagi taqqoslov tahlil shuni ko'rsatadiki, Flutter tezroq rendering, boy vizual interfeys va mustaqil grafik dvijok bilan ajralib turadi, React Native esa platformaga xos komponentlar orqali yanada tabiiyroq foydalanuvchi tajribasini taqdim etadi. Flutter'da tayyor widgetlar ko'pligi sababli murakkab interfeyslar tezroq quriladi, React Native esa mavjud kutubxonalar va uchinchi tomon yechimlari orqali kengaytirilishi bilan ustunlik qiladi.

Ish unumdorligi nuqtai nazaridan, har ikkala texnologiya ham samarali bo‘lib, tanlov aniq loyiha talablariga bog‘liq holda belgilanadi. Masalan, vizual dizaynga boy, animatsion interfeyslar talab qilinadigan hollarda Flutter ustunroq bo‘lsa, tizim resurslari cheklangan va mahalliy funksiyalar bilan chuqur integratsiya zarur bo‘lgan holatlarda React Native maqsadga muvofiq bo‘ladi. Shunday qilib, har ikkala texnologiya o‘ziga xos imkoniyatlarga ega bo‘lib, ular zamonaviy dasturiy mahsulotlarni tez, samarali va sifatli ishlab chiqish uchun keng imkoniyatlar yaratmoqda.

Kross-platformali texnologiyalar asosida amaliy loyihalarni ishlab chiqish jarayoni ko‘p bosqichli va tizimli yondashuvni talab etadi. Ushbu jarayon asosan ehtiyojlarni aniqlash, dizayn, dasturlash, testlash va ishga tushirish bosqichlarini o‘z ichiga oladi. Dastlab, loyiha uchun qo‘yilgan funksional va texnik talablar aniqlanadi. Bu bosqichda foydalanuvchi auditoriyasi, foydalaniladigan qurilmalar, ishlatiladigan servislar va ma’lumotlar bazasi bilan integratsiya masalalari aniqlashtiriladi. Keyingi muhim qadam – foydalanuvchi interfeysini loyihalash, ya’ni UI/UX dizaynni ishlab chiqishdir. Kross-platformali loyihalarda dizayn yaratishda bir nechta platformalar uchun moslashuvchan, javob beruvchi (responsive) interfeyslar tayyorlanishi zarur. Flutter’da bu jarayon widgetlar yordamida bajarilsa, React Native’da komponentlar asosida amalga oshiriladi. Dizayn fazasi tugagach, asosiy dasturlash bosqichiga o‘tiladi. Bu yerda front-end interfeys komponentlari va backend bilan aloqani ta’minlovchi modullar ishlab chiqiladi. Ilova foydalanuvchiga xizmat ko‘rsatishi uchun API orqali ma’lumotlar bazasiga ulanadi. Ushbu integratsiya RESTful yoki GraphQL xizmatlar orqali amalga oshiriladi. Loyihaning muvaffaqiyati ko‘p jihatdan foydalanuvchi tajribasiga bog‘liq bo‘lib, bu esa optimallashtirilgan kod, silliq animatsiyalar va tez ishlov berish mexanizmlarini talab qiladi. Flutter’da setState, Provider, Bloc kabi state management vositalari, React Native’da esa Redux, Context API, MobX kabi texnologiyalar ishlatiladi. Ishlab chiqilgan ilova sinovdan o‘tkaziladi. Avval dasturchilar tomonidan funksional testlar, keyin esa real qurilmalarda foydalanuvchi tajribasi testlari o‘tkaziladi. Har ikkala texnologiya uchun ham testlashni avtomatlashtirish imkoniyati mavjud bo‘lib, bu loyiha sifatini oshiradi. So‘nggi bosqichda esa ilova App Store yoki Google Play kabi platformalarga joylashtiriladi, bu jarayonda zarur ruxsatnomalar, xavfsizlik sertifikatlar va foydalanuvchi siyosatlar ham qo‘shiladi. Umuman olganda, Flutter va React Native asosida loyihalar ishlab chiqish nafaqat ishlab chiqish xarajatlarini kamaytiradi, balki ilovani keng auditoriyaga tez yetkazish imkonini ham beradi. Shuningdek, bunday yondashuv talabalarni va yangi boshlovchi dasturchilarni murakkab platformalarga bog‘lanmasdan, yagona kod bazasi orqali amaliy mahsulot yaratishga undaydi, bu esa texnologik kompetensiyalarni oshirish va tajriba orttirishda muhim rol o‘ynaydi.

Tajriba natijalarini aniqlash maqsadida Flutter va React Native texnologiyalari

asosida ikkita turli xil mobil ilova prototipi ishlab chiqildi. Birinchi loyiha Flutter platformasida yaratilgan elektron kundalik ilovasi bo'lib, foydalanuvchilarga dars jadvali, baholar, eslatmalar va push xabarnomalar orqali maktab bilan bog'liq jarayonlarni boshqarish imkonini beradi. Ikkinchi loyiha esa React Native asosida ishlab chiqilgan sog'lom turmush tarzini nazorat qiluvchi mobil ilova bo'lib, foydalanuvchining kunlik harakatlari, suyuqlik iste'moli va sog'liq ko'rsatkichlarini qayd etib boradi. Har ikkala loyiha uchun yagona texnik talablar, dizayn sxemasi va ma'lumotlar bazasi strukturasi asosida ish olib borildi, bu esa texnologiyalar o'rtasidagi samaradorlikni qiyoslash imkonini berdi. Baholash mezonlari sifatida ilovaning ishga tushish tezligi, foydalanuvchi interfeysi javob berish vaqti, qurilma resurslaridan foydalanish darajasi (RAM, CPU), hamda foydalanuvchi tajribasiga oid subyektiv fikrlar asos qilib olindi. Flutter asosidagi ilova tez ishga tushishi, silliq animatsiyalar va vizual elementlarning aniqligi bilan ajralib turdi. React Native ilovasi esa foydalanuvchi interfeysining tabiiy ko'rinishi, kichik hajmdagi dastur fayllari va JavaScript'ga asoslangan kodning qayta foydalanish qulayligi bilan samaradorlik ko'rsatkichlarini namoyon etdi. Har ikkala ilova 20 nafar foydalanuvchi ishtirokida testdan o'tkazildi. Sinov davomida foydalanuvchilar ilovalarning funksiyalarini bajarishdagi qulaylik, tezlik va umumiy interfeysga nisbatan munosabatlarini bildirishdi. Fikrlar asosida aniqlanishicha, Flutter ilovasi dizayn jozibadorligi va tezkorligi bilan yuqori baholangan bo'lsa, React Native ilovasi esa intuitiv interfeys va kodni tez o'zgartirish imkoniyati bilan ijobiy fikrlar oldi. Shuningdek, tizim resurslarini tahlil qilishda Flutter ilovasi nisbatan ko'proq xotira talab qilgani, ammo barqarorlik darajasi yuqori ekani kuzatildi. Umuman olganda, tajriba natijalari shuni ko'rsatadiki, har ikkala texnologiya ham o'ziga xos afzalliklarga ega bo'lib, ularni tanlashda loyiha turlari, jamoaning texnik salohiyati va foydalanuvchi ehtiyojlari asosiy rol o'ynaydi. Bu tajriba orqali kross-platformali texnologiyalar yordamida amaliy loyihalarni muvaffaqiyatli ishlab chiqish imkoniyati mavjudligi isbotlandi.

O'tkazilgan tahlil va amaliy tajribalar asosida xulosa qilish mumkinki, kross-platformali dasturlash texnologiyalari — ayniqsa, Flutter va React Native — zamonaviy mobil ilovalar ishlab chiqishda samaradorlik, vaqt tejamlorligi va keng platformalarda foydalanish imkoniyati nuqtai nazaridan muhim o'rin egallaydi. Ushbu texnologiyalar yordamida ishlab chiqilgan ilovalar foydalanuvchi ehtiyojlarini to'laqonli qondirishi, tezkorlik, vizual silliqlik va foydalanishdagi qulaylik jihatidan yuqori ko'rsatkichlarga ega bo'lishi mumkin. Flutter texnologiyasi kuchli vizual interfeyslar, yuqori darajadagi animatsiyalar va bir xil dizayn tajribasini taqdim etsa, React Native esa soddaligi, keng JavaScript hamjamiyati va platformalarga xos interfeys komponentlari orqali afzallikka ega. Tajriba natijalari har ikkala yondashuv ham amaliy jihatdan foydali va ishonchli ekanligini ko'rsatdi. Shu bilan birga, har bir texnologiyaning o'ziga xos cheklavlari va moslashish xususiyatlari mavjud bo'lib, ular

loyiha spetsifikasiga qarab baholanishi lozim. Kelgusida kross-platformali dasturlash texnologiyalarini o'rganish va o'qitishda amaliy loyiha asosida o'rgatish, real tajribaga asoslangan yondashuvlarni tatbiq etish maqsadga muvofiqdir. Talabalarni real muammolar asosida ishlashga yo'naltirish nafaqat texnik ko'nikmalarni, balki muammoga yechim topish salohiyatini ham rivojlantiradi. Shuningdek, ushbu yo'nalishda ilmiy tadqiqotlar davom ettirilib, texnologiyalar o'rtasidagi chuqurroq taqqoslov, xavfsizlik, uzluksiz integratsiya va foydalanuvchi tajribasi kabi jihatlar bo'yicha yanada kengroq tahlillar olib borish tavsiya etiladi. Natijaviy jihatdan, kross-platformali dasturlash bugungi dasturiy mahsulotlar yaratishda strategik vositaga aylanib borayotgani va bu yo'nalish bo'yicha malakali mutaxassislar tayyorlash zaruriyati ortib borayotgani aniq namoyon bo'lmoqda.

FOYDALANILGAN ADABIYOTLAR:

1. Google Developers. (2024). Flutter documentation. <https://flutter.dev/docs>
2. Meta Platforms, Inc. (2024). React Native official documentation. <https://reactnative.dev>
3. Zamani, M., & Amin, A. (2022). Comparative analysis of cross-platform mobile development frameworks: Flutter vs. React Native. *International Journal of Computer Applications*, 184(12), 22-27.
4. Kumar, A., & Singh, R. (2021). Cross-Platform Mobile App Development Approaches: A Comparative Study. *Journal of Software Engineering and Applications*, 14(6), 289–297.